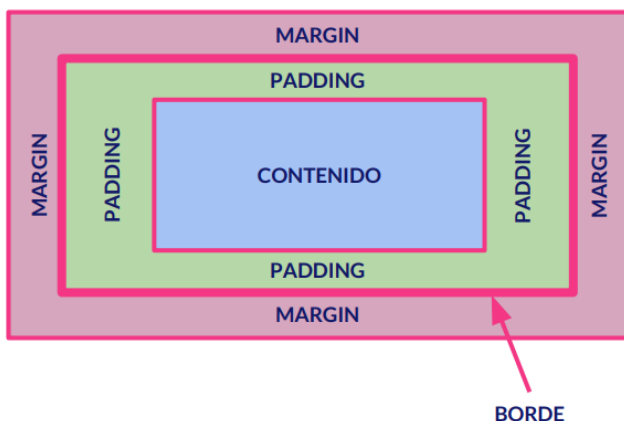


# Estructura amb CSS i Layouts

El model de caixa	1
Exemple caixa	2
Ojo amb el Width and Height	2
Display property	2
Modificar el Display value	3
Float property	4
Clearfix Hack	5
Centrar l'alineació dels elements	5
Center align elements	5
Center Align Text	5
Website Layout	5
With float	6
Width Grid	6
Width flexbox-bootstrap	6
Debat: quina és la millor manera de fer un layout?	6
Referencies	6

## El model de caixa

Tots els elements HTML es poden considerar com a quadres



- Contingut: el contingut del quadre, on apareixen text i imatges
- Padding: esborra una àrea al voltant del contingut. El farciment és transparent

- Border: una vora que envolta el farciment i el contingut
- Margin: neteja una àrea fora de la frontera. El marge és transparent

**i** El model de caixa ens permet afegir una vora al voltant dels elements i definir l'espai entre elements.

## Exemple caixa

```
<style>
  div {
    background-color: lightgrey;
    width: 320px;
    padding: 20px;
    border: 10px solid gray;
    margin: 0;
  }
</style>
```

## Ojo amb el Width and Height

Here is the calculation:

```
320px (width)
+ 20px (left + right padding)
+ 10px (left + right border)
+ 0px (left + right margin)
= 350px
```

The total width of an element should be calculated like this:

Total element width = width + left padding + right padding + left border + right border + left margin + right margin

The total height of an element should be calculated like this:

Total element height = height + top padding + bottom padding + top border + bottom border + top margin + bottom margin

## Display property

La propietat display és una altra propietat CSS per controlar el disseny.

[https://www.w3schools.com/css/css\\_display\\_visibility.asp](https://www.w3schools.com/css/css_display_visibility.asp)

És la propietat que especifica com visualitzem els elements.

Cada element HTML té un valor de visualització predeterminat en funció del tipus d'element que sigui. **El valor de visualització predeterminat per a la majoria dels elements és en bloc o en línia.**

- Caixa en línia - **display: inline**
- Caixa en línia i bloc **display: inline-block**
- Caixa en Bloc - **display: block**

## The display Property

### display: inline

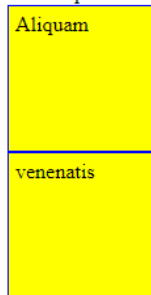
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum consequat scelerisque elit sit amet consequat. Aliquam erat volutpat. Aliquam venenatis gravida nisl sit amet facilisis. Nullam cursus fermentum velit sed laoreet.

### display: inline-block

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum consequat scelerisque elit sit amet consequat. Aliquam erat volutpat. Aliquam venenatis gravida nisl sit amet facilisis. Nullam cursus fermentum velit sed laoreet.

### display: block

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum consequat scelerisque elit sit amet consequat. Aliquam erat volutpat.



gravida nisl sit amet facilisis. Nullam cursus fermentum velit sed laoreet.

## Modificar el Display value

Tal com hem vist amb html cada element té un valor de visualització predeterminat.

Tanmateix, podem modificar-ho.

Canviar un element en línia a un element de bloc, o viceversa, pot a vegades ser útil. Un exemple comú és crear elements <li> en línia per als menús horitzontals:

```
<style>
    li {
        display: inline;
    }
</style>
<body>
    <p>Display a list of links as a horizontal menu:</p>
```

```

<ul>
  <li><a href="/html/default.asp" target="_blank">HTML</a></li>
  <li><a href="/css/default.asp" target="_blank">CSS</a></li>
  <li><a href="/js/default.asp" target="_blank">JavaScript</a></li>
</ul>
</body>

```



*Nota:*

Un **element en bloc** comença en una línia nova i ocupa tota l'amplada disponible

Exemples: <div>, <h1>, <p>, <ul>, <form>, <header>, <footer>, <section>

Un **element en línia** no comença en una línia nova i només ocupa l'amplada necessària. Exemples: <span>, <a>, <img>

## Float property

Amb la propietat float, és fàcil col·locar elements un al costat de l'altre:

- **La propietat flotant CSS** especifica com ha de flotar un element dins d'un container:
  - left - L'element flota a l'esquerra del seu contenidor.
  - right - L'element flota a la dreta del seu contenidor
  - none - **Per defecte** L'element no flota (es mostrarà just on apareix al text).
  - hereta: l'element hereta el valor flotant del seu pare
- **La propietat clear CSS** especifica quins elements poden flotar al costat de l'element esborrat i de quin costat.
  - none: Per defecte. l'element no es mou.
  - esquerra: l'element es mou per sota dels elements flotants esquerre
  - dreta: l'element es mou per sota dels elements flotants a la dreta
  - ambdós: l'element es mou per sota dels elements flotants esquerre i dret
  - hereta: l'element hereta el valor clar del seu pare.

Si un element flota cap a l'esquerra, hem de netejar cap a l'esquerra. L'element flotant continuarà flotant, però l'element esborrat apareixerà a sota a la pàgina web.

```

#nova {
  clear:left;
}

```



Quan utilitzem la propietat float i volem el següent element a continuació (no a la dreta ni a l'esquerra), haurem d'utilitzar la propietat clear. Fer exemple W3School: [https://www.w3schools.com/css/tryit.asp?filename=trycss\\_layout\\_clear](https://www.w3schools.com/css/tryit.asp?filename=trycss_layout_clear)

## Clearfix Hack

Enlloc d'anar fent clears... podem forçar un element a netejar els seus fills.

```
.clearfix:after {  
    content: "";  
    display: table;  
    clear: both;  
}
```



*TIP: com funciona el pseudoselector ::after*

[https://www.w3schools.com/cssref/tryit.asp?filename=trycss\\_sel\\_after\\_style](https://www.w3schools.com/cssref/tryit.asp?filename=trycss_sel_after_style)

## Centrar l'alineació dels elements

### Center align elements

Per centrar horitzontalment un element de bloc (com <div>), utilitzeu margin: auto;

Establir l'amplada de l'element evitarà que s'estiri fins a les vores del seu contenidor.

Aleshores, l'element ocuparà l'amplada especificada i l'espai restant es dividirà a parts iguals entre els dos marges

```
.center {  
    margin: auto;  
    width: 50%;  
    border: 3px solid green;  
    padding: 10px;  
}
```

[https://www.w3schools.com/css/css\\_align.asp](https://www.w3schools.com/css/css_align.asp)

### Center Align Text

```
.center {  
    text-align: center;  
    border: 3px solid green;  
}
```

Per centrar només el text dins d'un element, utilitzeu text-align: center;

## Website Layout

Un lloc web sovint es divideix en encapçalaments, menús, contingut i peu de plana



## With float

[https://www.w3schools.com/css/css\\_website\\_layout.asp](https://www.w3schools.com/css/css_website_layout.asp)

## Width Grid

[https://www.w3schools.com/css/css\\_grid.asp](https://www.w3schools.com/css/css_grid.asp)

## Width flexbox-bootstrap

La diferència més gran entre Bootstrap 3 i Bootstrap 4 i 5 és que Bootstrap 5 ara utilitza flexbox, en lloc de floats, per gestionar el disseny.

<https://getbootstrap.com/docs/5.0/layout/grid/>

## Debat: quina és la millor manera de fer un layout?

<https://www.youtube.com/watch?v=HHCRxrfcGLk> (\*)

<https://www.youtube.com/watch?v=UBoEXXzjUw4>

<https://www.youtube.com/watch?v=hYJvxsgnGMA>

## Referencies

[https://www.w3schools.com/css/css\\_float.asp](https://www.w3schools.com/css/css_float.asp)